Module1 final report

1. **(7,4) hamming code四種解碼方式討論與比較**

Channel decoding是恢復原始訊息經過雜訊通道傳輸的過程。(7,4) hamming code是一種廣泛使用的錯誤更正碼，可以更正傳輸數據中bit的錯誤。Module 1中使用了4種方法解碼Hamming code，包括Syndrome Decoding、Maximum Likelihood、Support Vector Machine和Deep Learning。以下將比較這些方法：

1. Syndrome Decoding：

Syndrome Decoding是一種常用的hamming code解碼方法。在這種方法中，先將接收到的經過雜訊的訊號做hard decision變成binary code。

接著將binary code與parity check matrix (H) 相乘為syndrome vector。

然後將syndrome vector與所有可能出現error pattrern表進行比較，更正相應的error以獲得原始訊號bit。

|  |  |
| --- | --- |
| Syndrome | Error pattern |
| (0, 0, 0) | (0, 0, 0, 0, 0, 0, 0) |
| (0, 0, 1) | (0, 0, 0, 0, 0, 0, 1) |
| (0, 1, 0) | (0, 0, 0, 0, 0, 1, 0) |
| (0, 1, 1) | (0, 0, 1, 0, 0, 0, 0) |
| (1, 0, 0) | (0, 0, 0, 0, 1, 0, 0) |
| (1, 0, 1) | (0, 1, 0, 0, 0, 0, 0) |
| (1, 1, 0) | (1, 0, 0, 0, 0, 0, 0) |
| (1, 1, 1) | (0, 0, 0, 1, 0, 0, 0) |

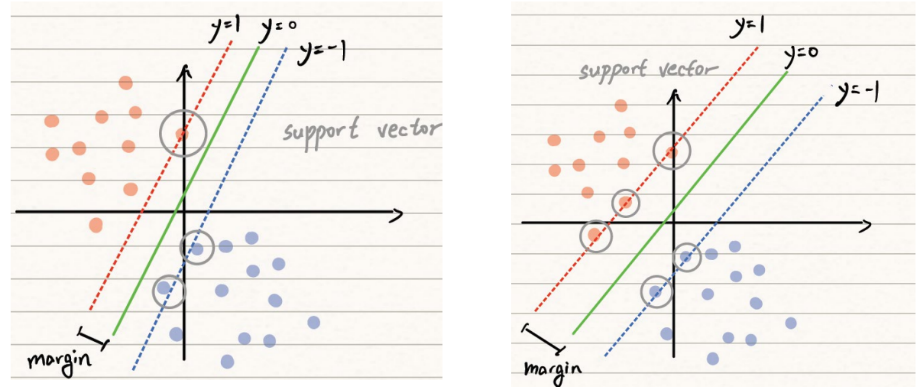
這種方法相對簡單高效，但因為只考慮最有可能出現的錯誤，若有多個錯誤時，判別的正確率會較低。

1. Maximum Likelihood：

Maximum Likelihood旨在將所有可能收到的訊息mapping到最有可能的原始訊息。在 (7,4) hamming code的情況下，可以透過計算接收到的code與所有可能的code之間的distance來實現。然後選擇距離最小的code作為發送code。

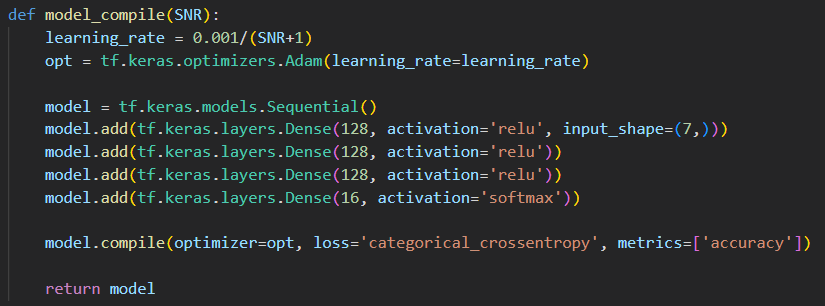
這種方法可以更正多個錯誤，可以達到最佳的解碼性能，但因為需要考慮所有可能性，所以計算較syndrome decoding複雜。

1. Support Vector Machine：

Support Vector Machine是一種監督式的學習方法，用統計風險最小化的原則來分類，其基礎就是找到一個決策邊界 (decision boundary) 讓兩類之間的邊界 (margins) 最大化，使其可以完美區隔開來。SVM用於解碼就是將接收到的code分類為特定codeword。

SVM需要預先用training data來訓練，因此沒有Syndrome decoding、Maximum Likelihood來得及時，也容易受data品質影響，不過若已知雜訊模型，這種方法便有良好的解碼性能。因為有考慮多個bit錯誤的情況，所以表現較Syndrome decoding好，但不會像maximum likelihood有高準確性。

Deep Learning：

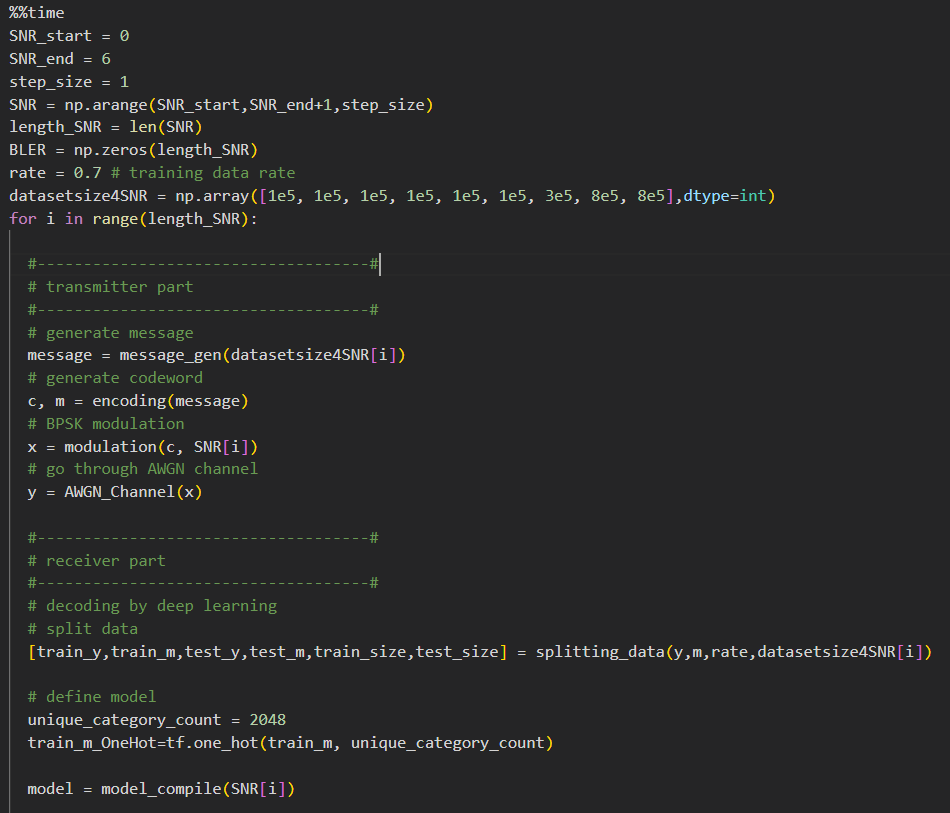
Deep Learning使用neural network來學習接收到的code和相應的傳輸code之間的對應關係。此方法需要給定neural network layer和nodes的數量，以及activate function、learning rate等等，並使用大量training data來訓練，可能非常耗時。

然而，若參數調適得當，此方法可以獲得良好的解碼性能並且可以很好地應用到新的data，表現可以接近Maximum likelihood。

總之，這四種方法都有其優點和局限性。Syndrome decoding簡單高效，但可能無法更正多個錯誤。Maximum likelihood是最佳解但計算複雜。Support vector machine和Deep learning是基於機器學習的方法，可以實現良好的解碼性能，但需要訓練並且可能無法很好地應用到新數據。方法的選擇取決於具體的應用要求，包括所需的解碼性能、計算複雜度和可用的training data。

1. **(15,11) hamming code程式說明**

在module 1的mini project中，將使用Deep Learning的方法解碼，不過這次原始訊息的編碼方法是使用(15, 11) hamming code。整個程式將與之前一樣比較SNR=0~6，過程從訊息產生、編碼、modulate，到加入雜訊，最後建立並訓練模型，解碼訊息。



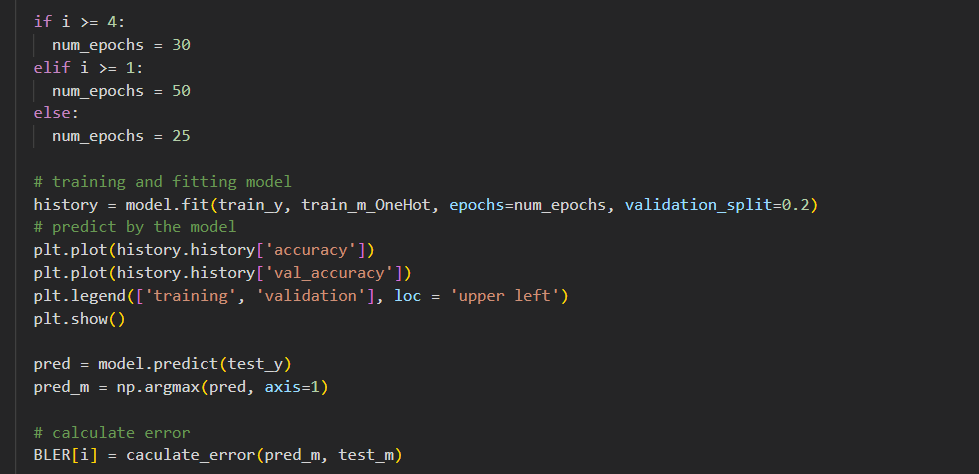
**產生訊息**

**用 (15,11) hamming code編碼**

**modulate**

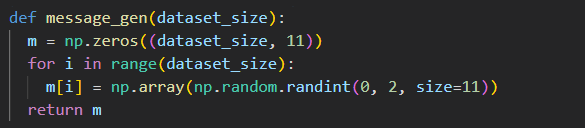
**加入雜訊**

**建立neural network**



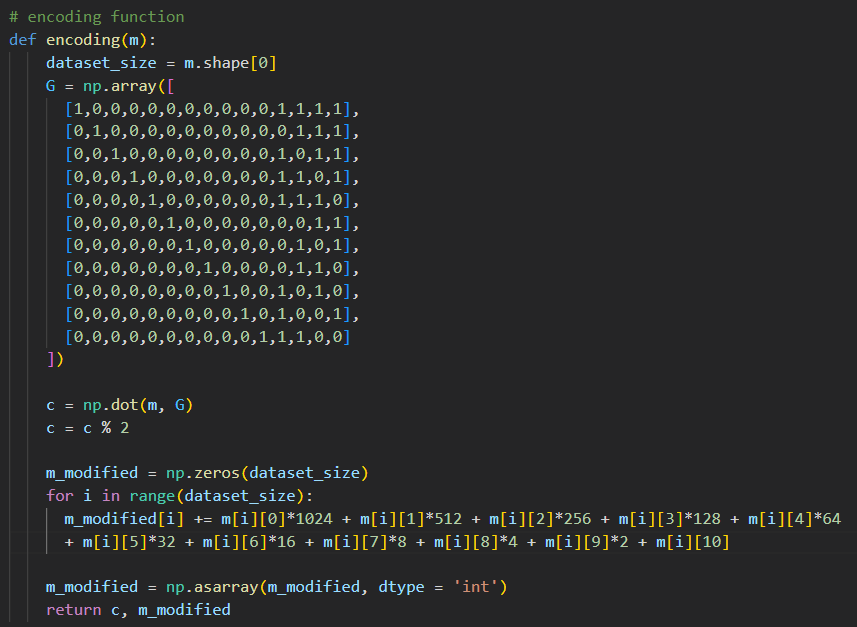
**訓練模型**

**預測 (解碼)**

* 1. 產生訊息:

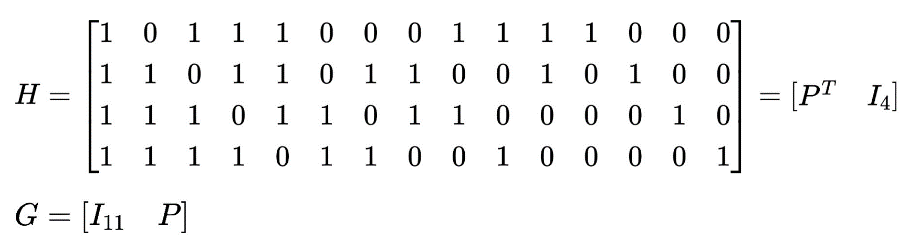
隨機產生11-bit的binary訊息。

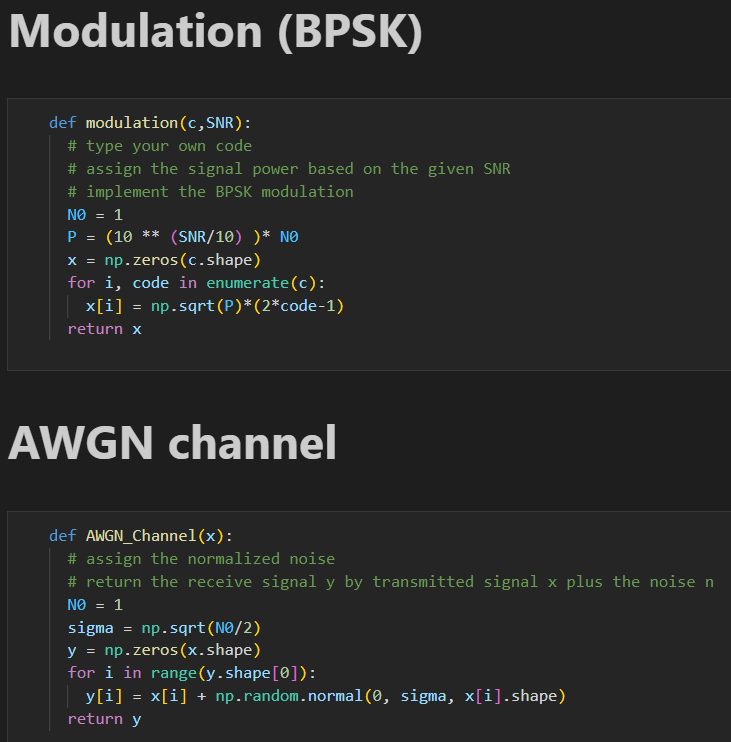
* 1. 用(15, 11) hamming code編碼:



**Generator matrix**

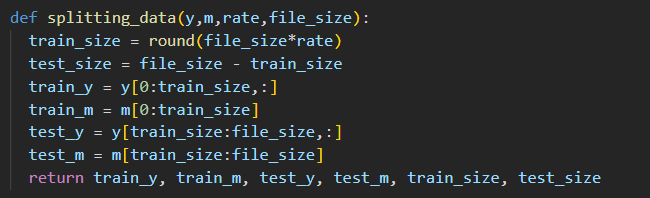
**轉換成decimal**

******將原始訊息乘上generator matrix，得到15-bit的編碼訊息 ()。這裡將11-bit的binary message轉換成decimal，以便之後one hot分類。

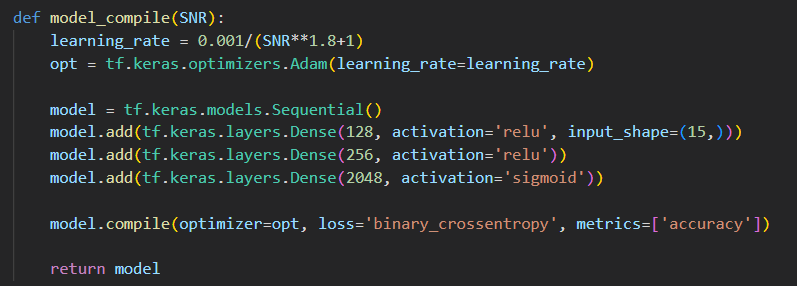
* 1. Modulate、加入雜訊:

將編碼過的message以對應的訊雜比 (SNR) 調變，並加上normal distribution的雜訊，變成接收到的訊號。

* 1. Split data:

將message和接收訊息分成training data和testing data。

* 1. 建立neural network:



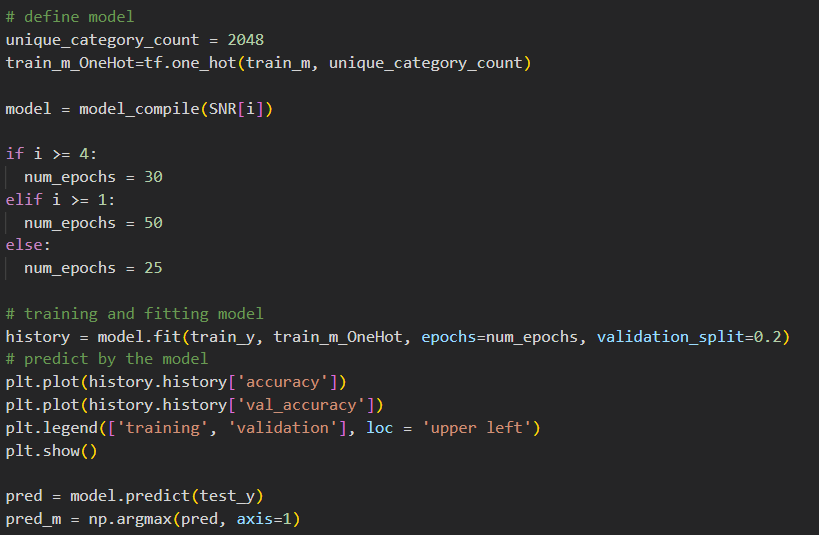
**不同layer**

**每層layer中node數量**

**Activate function**

給定layer、node，以及activate function，並根據SNR調整learning rate (SNR越大，模型越容易收斂，因此learning rate調小)。

* 1. 訓練模型並預測:



**211類**

**訓練模型**

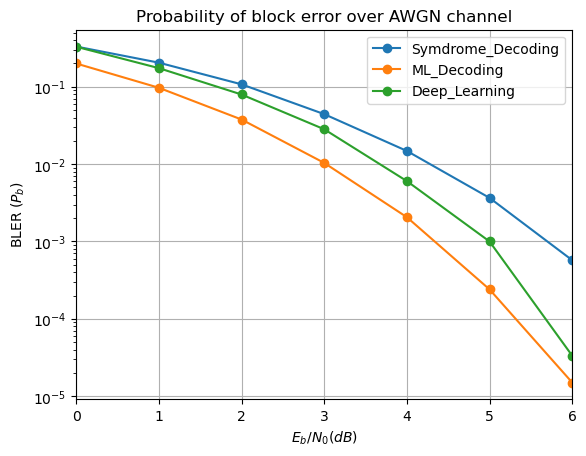
**預測 (解碼)**

將training data丟入模型訓練，因為message為11-bit，所以模型實際上是做211=2048類的分類。訓練完後丟入test data，得到預測(解碼)後的訊息。

* 1. 結果

訓練過程中，SNR越小，validation的曲線難接近100%，反之fold validation則能預測精準。

|  |  |
| --- | --- |
| SNR = 0 | SNR = 6 |

最後調參完的結果，表現有超過Syndrome decoding，雖然SNR小時與Syndrome decoding 差不多(learning rate應該可以再小一點)，不過到SNR=6時幾乎能接近Maximum likelihood。